A new public-key crypto system via Mersenne numbers

Divesh Aggarwal
joint work with Antoine Joux, Anupam Prakash and Miklos Santha

Public-key cryptography

-Introduced by Diffie and Hellman in [DH76]

-Many candidates over the years

-The quest in the recent years has shifted to advanced primitives

-In this work, we propose an arguably simpler PKC scheme.
-We also believe it is secure against quantum attacks.

Mersenne cryptosystem

- Belongs to the Ring and Noise family with
  - NTRU
  - Code-based crypto
  - Ring LWE based crypto
- With a different Ring: $\mathbb{Z}/p\mathbb{Z}$ (p Mersenne prime), and
- a different Noise: Hamming weight mod p.

Mersenne cryptosystem

Mersenne primes: They are primes of the form $p=2^n-1$, where n is a prime, and is named after Marin Mersenne, a French mathematician, who studied them in the early 17th century. (Wikipedia)

Main advantage of the cryptosystem: Simplicity

Mersenne ring and distance

- Ring Z/pZ
  - p a Mersenne prime, i.e., 2^n-1

Let :
- Rp(X)=rep of X in [0,p-1]
- HW(X)=num of 1 in binary rep of X mod p

Some properties of arithmetic mod p

1) HW(X+Y) ≤ HW(X) + HW(Y)

```
 1101010111001
+0000000001000
--------------
=1101010000001
```

2) For all i, HW(X 2^i) = HW(X)

3) HW(XY) ≤ HW(X) x HW(Y)
   Induction

4) HW(-X) = n-HW(X)

Warm Up

Single bit version

Hard problem

p = 2^n - 1, h << n

f, g are numbers mod p with few (< h) 1s in binary rep.

H=f/g [mod p]

Assumption: Given H, obtain f, g.

Single bit version

$H = f/g \pmod p$, PK = H, SK = g
(f and g containing few 1s, i.e. $\leq h$)

Encryption	Decryption
a and b with few 1s	$gC = \pm [a f + b g]$
$C0 = Enc(0) = (a H + b)$	Compute HW(gC)
$C1 = Enc(1) = -(a H + b)$	Small => 0
	Large => 1

Toy Example

$p = 2^{31}-1 = 2147483647 = 0x7FFFFFFF$
$H = f/g = 0x8002000/0x20000008$
$= 0x42E8BE0F$

Encryption	Decryption
$a = 0x80800$	
$b = 0x40000080$	$gC = 0x110084A6$
$C = Enc(0) = (a H + b)$	$HW(gC) = 8 (< 15) \Rightarrow 0$
$= 0x766CAB3A$	

Correctness of decryption

For correctness, we need $n > 4 h^2$

$g(aH+b) \equiv af+bg \pmod p$

$HW(Rp(af+bg)) \leq HW(a)HW(f)+HW(b)HW(g)$
$\leq 2 h^2 \leq n/2$

$HW(Rp(-(af+bg))) = n - HW(Rp(af+bg))$
$\geq n/2$

Multi-bit version

underlying encryption

Change public/private key

$H=f/g \pmod p \quad \Leftrightarrow \quad f(-1/H)+g=0 \pmod p$

I.e. $fR+g=0$

$T=fR+g \pmod p$ (R fully random)

Mersenne (basic multi-bit encrypt)

$T=fR+g \pmod p$ (R fully random)

Encryption	Decryption of (C1,Z)
$C1=aR+b1$	$C2'=fC1$
$C2=aT+b2$	
$Z=C2 \oplus E(m)$	$m=D(C2' \oplus Z)$
$Enc(m)=(C1, Z)$	

E and D : Error correction code

Multi-bit encryption

Analysis of decryption

$C2 = aT + b2 = afR + (ag+b2)$
$C2' = fC1 = f(aR + b1) = afR + b1\,f$

$HW(C2 \oplus C2') \le Hdist(C2,afR) + Hdist(C2',afR)$

Thus $Dec(Enc(m) \oplus$ small error$) = m$

Heuristic : Error is well distributed
Allows to use simple repetition code

Analysis of decryption

LEMMA: Let U be a random n-bit string and let x be an n-bit string of Hamming weight h. Then

$Pr[Hdist(U, U + x) > 2h(1 + c)] <$ negligible

EXAMPLE:

11001010101011110101110101000111110100101
+00001000000100010000000010001000010010

11010010101101111101110101101001111000111

Choice of error-correcting code

-Thus, the total number of errors we expect is at most $e = 2\,(2\,h^2 + h)$

-We need an ECC correcting e out of n errors

-Can use Reed Muller codes, and $n = O(h^2)$

-The number e is clearly an overestimate of the no. of errors in practice

-Also, we expect the errors to be distributed randomly

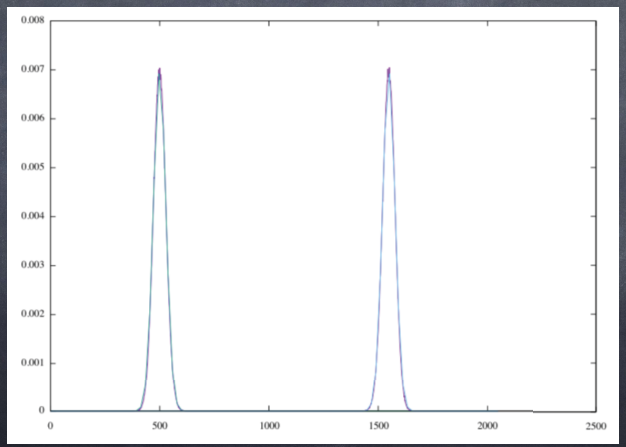
Recommended parameters

$n = 756839$

Low HW parameter $h=256$

Encode 256 bits:
with 2048-repetition coding

Heuristics



Hard Problem

Distinguish

Hidden low weight

$(R1, R2, aR1+b1, a\,R2+b2)$

$a, b1, b2$ with low HW

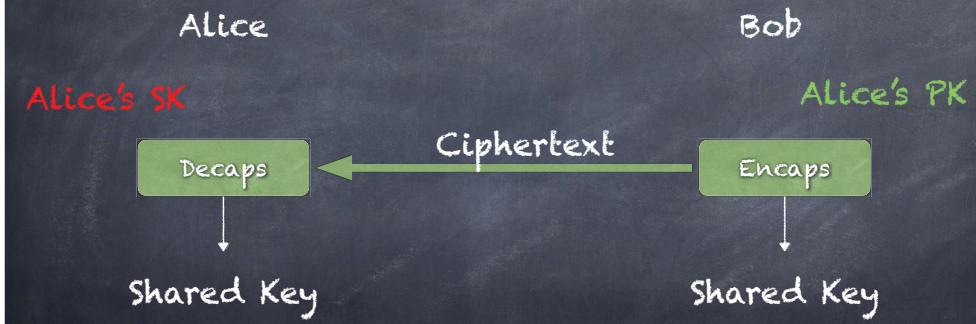
Random tuple

$(R1, R2, R3, R4)$

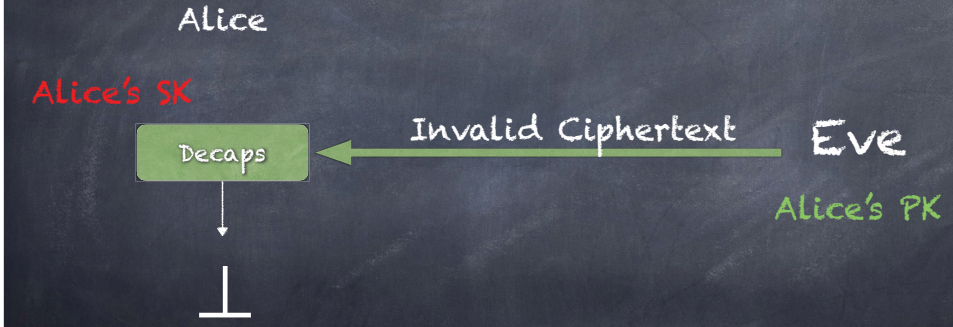
Multi-bit Mersenne

CCA-KEM

CCA-KEM



CCA-KEM under active attack



Mersenne KEM encaps (with CCA security)

s = Random seed

1) Initialize PRG from s
2) Produce pseudo random shared secret
3) Run basic encryption of s
   (getting a, b1, b2 from PRG)
4) Output (C1, Z)

Mersenne KEM decaps (with CCA security)

1) Run basic decryption on (C1, Z)
2) Re-encapsulate from s
3) Compare and Output
   a) Shared secret
   b) or $\perp$

Best Known attacks [BCGN17, BDJW18] (for proposed params)

Trivial : $\binom{n}{h}$

Best Classical : At least 2^{2h}

Best Quantum : At least 2^{h}

Future Work

-Cryptanalysis

-Improve efficiency without compromising security

Thank You